Center
for
Economic Research

No. 2001-10

OPTIMIZATION VERSUS ROBUSTNESS IN SIMULATION: A PRACTICAL METHODOLOGY, WITH A PRODUCTION-MANAGEMENT CASE-STUDY

By Jack P.C. Kleijnen and Eric Gaury

February 2001

ISSN 0924-7815

C:\Data\WP\PAPERS\Robustness\RobusOR6.WPD

Printed: March 22, 2001 (9:12pm)

Version 6: March 22, 2001

Optimization versus robustness in simulation: a practical methodology, with a production-management case-study

Jack P.C. Kleijnen [a] **& Eric Gaury** [b]

[a] Department of Information Systems (BIK)/Center for Economic Research (CentER), Tilburg University (KUB), Postbox 90153, 5000 LE Tilburg, Netherlands

Phone: +3113-4662029; Fax: +3113-4663377; E-mail: kleijnen@kub.nl

Web: center.kub.nl/staff/kleijnen

[b] BIK/CentER, KUB and LIMOS/IFMA, Clermont Ferrand, France

Current address: Technocentre Renault, API: TCR AVA 3 61

1, Avenue du Golf, 78288 Guyancourt Cedex, FRANCE

Phone: +01 34 95 52 03; Fax: +01 34 95 56 25; E-mail: eric.gaury@renault.com

Keywords & descriptive phrases:

Design of experiments, statistical analysis, applications

In practice, a robust solution is more appealing than an optimal solution.

The methodology adds risk analysis and bootstrapping.

The case study concerns pull production-control systems.

Abstract

Whereas Operations Research has always paid much attention to optimization, practitioners judge the robustness of the 'optimum' solution to be of greater importance. Therefore this paper proposes a practical methodology that is a stagewise combination of the following four proven techniques: (1) discrete-event simulation, (2) heuristic optimization, (3) risk or uncertainty analysis, and (4) bootstrapping. This methodology is illustrated through a case study on production control systems. That study defines robustness as the system's capability to maintain a short-term service measure, in a variety of environments (scenarios). More precisely, this measure is the probability of the short-term fill rate remaining within a prespecified range. Besides satisfying this probabilistic constraint, the system should minimize long-term work-in-process. Actually, the case study compares four systems: Kanban, Conwip, Hybrid, and Generic. These systems are studied for a well-known example, namely a production line with four stations and a single product. The conclusion of this case study is that Hybrid is best when risk is not ignored, but otherwise Generic is best: risk considerations do make a difference.

1. Introduction

Operations Research (OR) has always paid much attention to *optimization*, as witnessed by the history of the Economic Order Quantity, Linear Programming, etc. More specifically, in simulation, both academic researchers and software developers have added optimization capabilities. For example, the 2000 Winter Simulation Conference had a panel on simulation optimization, in which participated eight experts from academia and software companies; see Fu (2000). However, during the panel's Question & Answer the speakers agreed that *robust-*

ness is of great importance in practice: a solution that is (nearly) optimal for a given scenario, is not practically relevant if that solution breaks down as soon as the environment changes. Nevertheless, commercial simulation packages do not provide the required robustness capabilities; also see Law and McComas (2000) and Swisher et al. (2000).

Because practitioners judge the robustness of the 'optimum' solution to be of utmost importance, we propose a practical *methodology* that is a stagewise combination of the following four well-known techniques: (1) discrete-event simulation, (2) heuristic optimization, (3) risk or uncertainty analysis, and (4) bootstrapping. (The famous economist Schumpeter spoke of 'innovations' in case of new combinations of old techniques. Neither risk analysis nor bootstrapping have ever before been used in production management - to the best of our knowledge.)

We illustrate this methodology through a *case study* on the robustness or riskiness of production management systems; that is, we examine the systems' sensitivity to changes in the assumed environment. We focus on *pull production-control systems* (PPCSs), namely Kanban systems and their variants. Our elaborate survey of the PPCS literature (see the many references at the end of this article) confirms that in this domain - as in many other OR domains - the analysts optimize, assuming a *specific* environment (scenario, combination of non-controllable input values). In practice, however, the future environment is unknown; for example, breakdown rates are unknown. Consequently, PPCS performance may be far below the manager's target. In practice, PPCS robustness is indeed judged to be a major issue (our personal contacts with managers support this statement).

More specifically we choose the example of a stochastic production line with four stations and a single product, from Bonvik, Couch, and Gershwin (1997). We compare the performance of this production line, under four optimized PPCSs - namely, Kanban, Conwip, Hybrid,

and Generic (defined in §3). Our methodology results in a performance ranking that indeed differs from the ranking resulting from other methods that ignore risk.

Note that similar problems are addressed by Mulvey, Vanderbei, and Zenios (1995), using robust linear programming including penalty functions and a set of scenarios with a probability distribution. Mulvey et al. (1995, p. 269) also comment on an alternative approach, namely stochastic linear programming. Interesting monographs on stochastic programming are Birge and Louveaux (19..) and Kall and Wallace (1994). Further, Ben-Tal (2000) also discusses robust convex optimization (including the optimally designed bridge that collapses as soon as a bird lands on it; also see Mulvey et al., 1995, pp. 276-277). However, none of these approaches can be applied straightforwardly to our discrete-event simulation models (which are inherently stochastic and dynamic), with their particular managerial criteria (which are more complicated than the criteria in Mulvey et al. 1995).

Note further that we do not consider a change of the number of kanbans, once the uncertain environmental variables are observed; see Mulvey et al. (1995, p. 265) and the dynamic control literature. One reason for our approach is that the true values of the parameters - such as the breakdown rates - never become certain.

Note finally that Gaury (2000) explores the relationships between robustness and the Taguchian viewpoint. Taguchians assume either quadratic loss functions or simple signal-noise functions that combine the mean and variance of the output; see Myers (1999). We, however, use performance measures that make sense from a managerial point of view. Taguchi's approach is applied in a Kanban study by Moeeni, Sanchez, and Vakharia(1997). That approach has also been used for job shops (Benjamin, Erraguntla, and Mayer 1995), and for production planning outside PPCSs (Lim, Kim, Yum, and Hwang 1996).

The remainder of this article is organized as follows. §2 explains our methodology in detail.

§3 illustrates this methodology through Bonvik et al.'s example. §4 summarizes our conclusions.

2. New methodology

In this section we detail our methodology that consists of five stages. As we go along, we illustrate our method through our case study (more details of this study will follow in the next section, §3).

Stage 1: We build a discrete-event simulation of the system. This is a classic technique, which needs no further comments or references.

In our case study the real system consists of the production system and its PPCS. This 'real' production system is the system in Bonvik et al. (1997). Sample output of our Arena simulation is shown in Figure 1 (discussed in §3).

Note that in practice, the simulation's distribution types and their parameter values are not exactly known, so sensitivity analysis is needed to test and improve the validity of the simulation model; see Kleijnen (1998). However, this sensitivity analysis plays no role in our study because it concerns only an academic example (we did verify some of our results against Bonvik et al.'s results).

Stage 2: Assuming a specific combination of non-controllable input values (e.g., the most likely environmental scenario), we try to find the best combination of controllable input values for the simulated system of stage 1. This optimization is challenging, since this simulation is stochastic, non-linear, and multi-response. This is a well-known issue; no solution technique is known to be superior (see the references in §1).

In our case study, we consider four PPCS types. Each type requires optimization (namely,

quantification of the number of cards or kanbans per control loop; each PPCS type has different control loops; see §3). We use a Genetic Algorithm (GA) combined with Response Surface Methodology (RSM); see Gaury (2000) and Gaury et al. (1999). However, other optimization techniques might be applied, as part of our methodology.

Stage 3: Apply risk analysis (RA) to estimate the probability of a specific system performance (output). RA uses the Monte Carlo method (i.e., random numbers) to sample from an assumed distribution of environmental inputs. RA is also called risk assessment, risk management, or uncertainty analysis; see Granger Morgan and Henrion (1990).

We feed this RA sample into the 'optimized' simulation model that results from stages 1 and 2. More specifically, our RA consists of the following three steps.

(3a) First we sample a value for each environmental input variable (e.g., the demand rate) from its input distribution; for simplicity we suppose that the inputs are independent. To increase the accuracy (i.e., reduce statistical variation), we do not use crude Monte Carlo but *Latin hypercube sampling* (LHS); LHS gives better coverage of the total sample space. LHS is a standard option in the @Risk software, which we use. Also see McKay, Beckman, and Conover (1979), and also Helton (1997).

(3b) Next we feed these sampled RA input values into the 'optimized' simulation model that resulted from stage 2. We run this simulation model to obtain one new realization of the (multiple) performance measures. Because we wish to minimize computer time, we do not replicate the simulation run for a particular scenario.

(3c) To estimate the distribution of the outputs, we repeat the RA steps (3a) and (3b) a number of times; this number is the *LHS sample size* (say) n. All these n runs with the simulation model start with the same initial conditions. In the case study we use different random numbers from run to run (common random numbers would have been an option). The resulting

distribution may be characterized by a single statistic such as the average or a specific quantile.

Note that RA is a standard technique in nuclear engineering (Helton et al. 1997, and also Balson, Welsh, and Wilson 1992 and Breeding et al. 1992). Recently, RA has become accepted by management too, because of the widespread availability of software - such as @Risk, Crystal Ball, and XLSim - that supplements popular spreadsheet programs (Sugiyama and Chow 1997). However, RA has been applied mainly to investment analysis (see Krumm and Rolle 1992) - not to PPCSs.

Stage 4: To estimate a (say) 90% confidence region for the performance measures, we apply bootstrapping. The bootstrap is a resampling technique, using Monte Carlo. We do not know of many bootstrap applications in OR, but in mathematical statistics it is a well-known technique. The seminal book on bootstrapping (outside simulation and RA) is Efron and Tibshirani (1993).

Note that bootstrapping in simulation raises an interesting question: instead of using the computer to generate responses through bootstrapping, the computer may be used to generate more simulation responses. In practice, however, replicating a simulation generally requires much more computer time than bootstrapping a simulation.

Stage 5: Let the managers select a particular control system - for example, a specific PPCS - that fits their specific risk attitude.

Risk management may be further supported as follows. Once we have finished the RA (stage 3), we try to identify the *important environmental inputs*. In RA it is customary to make scatter plots per input. Details on the statistical analysis of such scatter plots are given in Kleijnen and Helton (1999).

3. Case study: pull production-control systems

Our example concerns an example taken from Bonvik et al. (1997); we do not expect this particular example to favor our methodology. Bonvik et al. use the following assumptions.

Delivery of raw materials is continuous and infinite. Movements of products and cards are instantaneous. Inventory value is constant over the production line (value added ignored). Processing times at each station are lognormal with a mean of 0.98 (minutes) and a standard deviation of 0.02. Machines have times between failures and repair times that are exponentially distributed with means of 1,000 and 3 respectively. Demand interarrival time is constant, namely 1. If no finished product is available, then demand is lost; so it is essential to have a fill rate close to 100%. Actually the *fill-rate target* is 99.9%.

Bonvik et al. consider Kanban, Conwip, and Hybrid (besides two more systems that we do not examine); we add Generic. The first three PPCSs have already been discussed extensively in the literature, so now it suffices to characterize them as follows. *Kanban* has control loops that connect each production stage with its immediate predecessor. *Conwip* has a single loop, from the final to the initial production stage (see Spearman, Woodruff, and Hopp 1990). *Hybrid* simply combines Kanban and Conwip (see Bonvik et al. 1997). *Generic* is a general PPCS introduced by Gaury, Pierreval, and Kleijnen (2000). In principle, Generic connects each stage with all its predecessors; hence the three other PPCSs as special cases. Actually, Generic does not implement loops with non-restrictive card numbers; for example, Generic reduces to Conwip if the only restrictive loop - given the card numbers of the other loops - is the one that connects the last stage with the first stage.

Stage 1: First we develop simulation models for Bonvik et al.'s example. We use the following mathematical notation. Upper case letters denote random variables, lower case letters denote realizations of random variables and deterministic variables, and Greek letters

represent parameters to be estimated. We do not explicitly show the dependence on the PPCS type. We define

$\mu = E(\overline{W})$: expected average WIP ($\mu \geq 0$ because $\overline{W} \geq 0$);

Y: fill rate per shift ($0 \leq y \leq 1$; percentage of demand per work shift, satisfied from stock);

$\pi = P(Y < c_y)$: probability of Y dropping below a prespecified managerial threshold (say) c_y.

We speak of a *disaster* whenever y drops below c_y. We measure the PPCS's *short-term* performance by π, and its *long-term* performance by μ.

To *estimate* these two measures, we use discrete-event simulation. Our simulation model produces the following two (autocorrelated) time series:

w_t: WIP realized at simulated (continuous) time t;

y_i: fill rate realized in shift i.

To generate these time series, we need to decide on the simulation's initialization and termination. We chose a warming-up period of three days, with each working day having 900 minutes (15 hours) and two shifts per day. We stop after one simulated month with 22 working days: $0 \leq t \leq 19800$ and $i = 1, ..., 44$. An example of the simulated time series is given in Figure 1.

Note that Bonvik et al. use a longer runlength of 240,000 simulated time units, of which the first 9,600 time units are estimated to show transient behavior so statistics collected during this transient period are discarded. Our RA, however, uses a shorter runlength of 19,800 minutes, plus a start-up period of 2,700 minutes (these shorter runs will turn out to be acceptable; see below).

INSERT Figure 1: Simulated w_t (WIP at time t) and y_i (fill rate of shift i)

These time series w_t and y_i give the following estimates for the two performance measures:

$$\hat{\mu} = \int_{2700}^{2700 + 19800} w_t \, dt/19800;$$

$$\hat{\pi} = \sum_{i=1}^{44} I(y_i < c_y)/44 \text{ with indicator function } I(\ ).$$

This definition implies that it is worse to have (say) two shifts each with one lost sale than one shift with two lost sales. We use such a definition because we assume that it is psychologically worse for the manager to underperform twice.

Figure 2 gives an example of the estimated density function and the corresponding (cumulative) distribution of *Y*, and the resulting estimated disaster probability $\hat{\pi} = 0.455$ for $c_y = 0.95$.

INSERT Figure 2: Estimated distribution of *Y* (fill rate per shift) and disaster probability $\pi = P_y(Y < 0.95)$: a simulation example

Stage 2: We optimize each PPCS under the base scenario specified by Bonvik et al.'s assumptions. For this optimization, we use Bonvik et al.'s criteria: satisfy the prespecified fill-rate target while minimizing WIP. This gives the following numerical results.

After an exhaustive search, Bonvik et al. estimated the optimal card numbers to be: 15 for Conwip; 2, 2, 4, and 10 for Kanban; 15 and 2, 3, 5, and 15 for Hybrid (with the first 15 for the Conwip loop, etc.). Through a GA combined with RSM we find for Generic: 14 cards for the Conwip loop, 6 cards for stage 1, 3 cards for stage 2, and non-restrictive card numbers for all other loops so the latter loops are not implemented.

Under Bonvik et al.'s criteria, their 'optimized' Hybrid outperforms Kanban. Conwip's performance is between Kanban's and Hybrid's. Our Generic performs slightly better than Hybrid.

Stage 3: In RA we need additional robustness criteria, besides Bonvik et al.'s criteria. We emphasize that our methodology can be easily adapted to different managerial robustness criteria.

Traditionally, analysts focus on long-run, steady-state performance metrics. We, however, also consider the *short run*: If the managers' performance is bad in the short run, they will be fired - at least such performance is not good for their careers.

To define our robustness criteria, we add a subscript (say) *s* to all symbols introduced in stage 1 that denotes the (environmental) *scenario*; that is, in RA we repeat the simulation for different scenarios *S* with value *s*, which gives

$\mu_s = E(W|\ S = s)$;

$Y_s = (Y|\ S = s)$;

$\pi_s = P(Y < c_y|\ S = s) = P(Y_s < c_y)$.

In other words, in RA the two performance measures are random variables, because the scenarios are treated as random input variables. (Bayesians always have such a world view. Note that, even if the simulation model were deterministic, RA would still give random output.) So, by definition, these measures have a joint statistical distribution function.

This distribution function might be used by management to select a PPCS (see Figure 7, discussed below). We, however, think that it is more practical to characterize each of the two marginal functions through a single number. More precisely, we characterize the estimated marginal distribution of the estimated average WIP through its *average* (say) $\overline{\hat{\mu}}$. The estimated marginal distribution of the estimated disaster probability $\hat{\pi}$, however, we characterize through the estimated probability of this $\hat{\pi}$ being higher than another managerial threshold (say) c_π; this single number is the *probability* (say) $\hat{\rho}$. We shall give results for $c_\pi = 0.9$. The latter number implies that we assume that the managers have a risk-averse attitude: they wish to

avoid high probabilities of high disaster probabilities. (Again, our methodology also accepts other robustness criteria.) In summary, we define two robustness performance measures:

$\eta = \underset{s}{E}(\mathrm{M}_s) = \underset{s}{E}\,[E(W_s \mid S = s)]$: mean average WIP averaged over all scenarios (M has realizations μ);

ρ: probability of Π_s exceeding the managerial threshold c_π, under various scenarios.

To estimate this η, we use RA with a given input distribution of scenarios, resulting in the following estimate:

$\hat{\bar{\mu}} = \sum_{s=1}^{n} \hat{\mu}_s/n$: average WIP (in simulation) averaged over the n scenarios actually sampled (in RA).

Actually, we should replace μ by the capital letter M to denote the random character of this estimate $\hat{\bar{\mu}}$. (To estimate the randomness of $\hat{\bar{\mu}}$, we use bootstrapping; see stage 4 below)

Analogously, our RA estimate of ρ is

$\hat{\rho} = \sum_{s=1}^{n} I(\hat{\pi}_s \geq c_\pi)/n$: fraction of $\hat{\pi}_s$ that exceeds c_π in RA.

The challenge is to meet the constraint on the short-term fill-rate (see $\hat{\rho}$), at minimal long-term WIP (see $\hat{\bar{\mu}}$).

In our example we have no information on the likelihood of the various scenarios, so we assume that all scenarios are equally likely. Hence we use a uniform prior distribution per environmental input, and assume independent inputs. We consider the following 17 inputs: the processing time's mean and variance, MTBF (mean time between failures) and MTR (mean time to repair) per production stage, and the demand rate. In our RA we vary these 17 inputs over a range of ±5% around their base values.

An illustration is Figure 3, which concerns Kanban optimized for the base scenario, given a fill-rate threshold of 97% (c_y = 0.97). Part (a) shows the estimated marginal density function of the estimated disaster probability $\hat{\Pi}$ (with values $\hat{\pi}$); part (b) does the same for the other

criterion $\hat{M}$ (average WIP with values $\hat{\mu}$); part (c) gives the scatter plot that indicates the joint distribution of these two estimators.

INSERT Figure 3: Estimated density function of estimated disaster probability $\hat{\Pi}$ and average WIP $\hat{M}$ - for Kanban, optimized given 97% fill rate target: Bonvik et al's four-stage example

Figure 3 enables us to estimate the two robustness measures η and ρ through $\overline{\hat{\mu}}$ and $\hat{\rho}$ defined above. We emphasize that for different target values c_y and c_π the simulation does not need to be run again: Figures 2 and 3(a) demonstrate that the basic information is available to compute $\hat{\pi}$ and $\hat{\rho}$ for different target values.

This figure gives quite surprising results, we think. The disaster density function turns out to have a *bathtub* shape. So, under many scenarios no disasters occur: left-hand side in part a), at $\hat{\pi} = 0$. In these scenarios there is ample line capacity. Under many other scenarios the optimized Kanban system never gives the target fill rate of 97%: right hand, at $\hat{\pi} = 1$. In the latter scenarios there is lack of capacity. Part c shows that - unlike we conjectured - low disaster probabilities do not necessarily go together with high WIPs; actually, the coefficient of determination R^2 is only 0.01 (computed from $n = 100$ points). One explanation is that some scenarios give a maximum disaster probability of one, whatever the WIP is: the production system does not have enough capacity to satisfy demand.

Moreover, we try to identify the important environmental inputs through scatter plots per input. Two examples are given in Figure 4. We find that in Conwip the most important parameter (with first-order and higher-order effects) is the demand rate: part (a) suggests that a low demand interarrival time increases the disaster probability. Part (b) indicates that changes in the average processing time at the last production stage do *not* have a systematic effect on the

disaster probability. Part (a) makes sense: high demand tends to decrease the fill rate (this conclusion support our model's credibility).

INSERT Figure 4: Scatter plot of (a) an important, and (b) an unimportant input parameter in RA of Conwip

We illustrate some more characteristics of the PPCSs - through Figures 5 and 6. Figure 5 shows the estimated disaster probability $\hat{\pi}$ when we change the *number of cards* in Conwip. The optimal number of cards under the base scenario is 15 (computed in stage 2). Of course, the probability of zero disaster probability is highest when the number of cards is largest: see c = 50 at the left-hand side. Nevertheless, even with this number of cards, 18 out of 100 scenarios lead to a disaster probability of 1: see the right-hand side.

INSERT Figure 5: Effect of number of cards c on estimated disaster probability $\hat{\pi}$, in Conwip, estimated from n = 100 scenarios

Figure 6 shows the estimated 'disaster' probability - for several fill-rate *target values* c_y, namely 95%, 97%, and 99.9%. Obviously, the lower the threshold is, the higher is the probability of no 'disaster': see the left-hand side of the figure.

INSERT Figure 6: Effect of fill-rate target c_y on estimated disaster probability $\hat{\pi}$

Figures 3 and 5 have already illustrated the bathtub shape of the estimated density function of the estimated disaster probability $\hat{\pi}$ in Kanban and Conwip respectively. However, to

compare the four PPCSs, we prefer the *cumulated* density functions; see Figure 7 (which uses a fill-rate target value c_y of 0. 97). This figure shows that - whatever PPCS is used - some scenarios certainly give a disaster: see the right-hand side. (Actually, most disaster scenarios are characterized by mean demands that exceed production rates; see again Figure 4.)

INSERT Figure 7: Estimated density function of estimated disaster probability $\hat{\pi}$ and average WIP $\hat{\mu}$ for four PPCSs

As we said before, the distribution functions in Figure 7 might enable management to select a PPCS but we prefer to characterize each function through *a single number*: the average $\overline{\hat{\mu}}$ for WIP and the probability $\hat{\rho}$ for fill rate (with threshold $c_\pi = 0.9$).

Stage 4: We apply bootstrapping to estimate a 90% confidence region for the two performance measures. A particular scenario *s* - in the LHS sample of size *n* = 100 - gives the so-called 'original' multivariate output (say) $\boldsymbol{x}_s = (\hat{\mu}_s, \hat{\pi}_s)$. Bootstrapping means that this original sample is resampled randomly with replacement, while the total sample remains *n* = 100. This gives the bootstrap output $\boldsymbol{x}_s^* = (\hat{\mu}_s^*, \hat{\pi}_s^*)$ (*s* = 1, ..., *n*). In other words, the original output for (say) scenario 1 - denoted by $\boldsymbol{x}_1 = (\hat{\mu}_1, \hat{\pi}_1)$ - may occur 0, 1, ..., or *n* times in the bootstrap sample, provided the total sample size remains *n* (for example, the event '$\boldsymbol{x}_1$ occurs *n* times in a particular bootstrap sample' implies that the other *n* - 1 observations $\boldsymbol{x}_2$, ..., $\boldsymbol{x}_n$ do not occur at all in this sample - a very unlikely event, namely an event with probability n^{-n}). From the bootstrap sample $(\hat{\mu}_1^*, \hat{\pi}_1^*)$, ..., $(\hat{\mu}_s^*, \hat{\pi}_s^*)$, ..., $(\hat{\mu}_n^*, \hat{\pi}_n^*)$ we compute the average $\overline{\hat{\mu}^*}$ and the probability $\hat{\rho}^*$. To estimate the distribution of these two criteria $(\overline{\hat{\mu}^*}, \hat{\rho}^*)$, we repeat this bootstrap sampling (say) *b* times; we select *b* = 200. This gives $(\overline{\hat{\mu}_j^*}, \hat{\rho}_j^*)$ with *j* = 1, ..., *b*; also see Figure 8.

INSERT Figure 8: Bootstrapped joint density function of the two robustness criteria ($\overline{\hat{\mu}}$, $\hat{\rho}$) for Generic

We wish to estimate a 90% simultaneous confidence region for the two estimated robustness criteria of a specific PPCS. Therefore we hypothesize that the bootstrapped variables are *bivariate normal*. To test this hypothesis, we apply Johnson and Wichern (1992, pp. 158-164), as follows. We denote the sampled multi-variate observations by X_j with $j = 1, ..., b$; in our example $x = (\overline{\hat{\mu}^*}, \hat{\rho}^*)$ and $b = 200$. We define the squared generalized distance as

$$D_j^2 = (\boldsymbol{X}_j - \overline{\boldsymbol{X}})' \boldsymbol{S}^{-1} (\boldsymbol{X}_j - \overline{\boldsymbol{X}}) \text{ with } j = 1, ..., b \qquad (1)$$

with bold letters for matrices and vectors, and the classic estimators $\overline{\boldsymbol{X}} = \sum_{j=1}^{b} \boldsymbol{X}_j / b$ and $\boldsymbol{S} = \sum_{j=1}^{b} (\boldsymbol{X}_j - \overline{\boldsymbol{X}})(\boldsymbol{X}_j - \overline{\boldsymbol{X}})' / (b - 1)$. Then the hypothesis of υ-variate normality (in our example $\upsilon = 2$) is not rejected if the following two conditions hold:

(i) roughly half of the d_j^2 are less than the 50% quantile of the chi-square statistic with υ degrees of freedom (say) $\chi_\upsilon^2(0.50)$, and

(ii) a plot of the b ordered d_j^2 versus the b quantiles $\chi_\upsilon^2([j - 0.5)/b])$ gives a straight line; see Figure 9, part c.

Visual inspection of the two upper parts of Figure 9 suggests that normality holds for the estimated marginal density functions of the two individual criteria (even for the estimated probability $\hat{\rho}$). The lowest part of this figure corresponds with the test defined in equation (1), and does not lead to rejection of the normality assumption for Generic. For simplicity's sake we do not test normality for the other three PPCSs, but simply assume that this assumption also holds for these PPCSs.

INSERT Figure 9: Testing normality of the bootstrapped $\overline{\hat{\mu}}$ and $\hat{\rho}$ for Generic

Next we apply Johnson and Wichern (1992, p. 189), to derive a 1 - confidence region for the two bootstrapped robustness criteria (say) = (η, ρ):

$$b(\overline{\boldsymbol{X}} -)'\boldsymbol{S}^{-1}(\overline{\boldsymbol{X}} -) \leq [2(b - 1)/(b - 2)]f_{2;\, b-2}(1 -) \qquad (2)$$

where $f_{2;\, b-2}(1 -)$ denotes the 1 - quantile (upper point) of the F-statistic with degrees of freedom 2 and b - 2.

We might apply equation (2) to each of the four PPCSs with a type-I error rate of α. However, our selection of a PPCS depends on all four confidence regions *simultaneously*. Therefore we use Bonferroni's inequality: we replace α by α /4, which keeps the overall type-I error rate below α. Taking α = 0.10 yields Figure 10. This figure shows that - even though Bonferroni's inequality is conservative - our example gives four non-overlapping confidence intervals for the WIP criterion $\overline{\hat{\mu}}$. However, the differences for the fill-rate criterion $\hat{\rho}$ are not significant. (Again, the two performance measures seem uncorrelated, as Figure 3 with its R^2 = 0.01 has already suggested: the ellipsoids in Figure 10 are not 'tilted' - that is, the estimated covariance matrix $\boldsymbol{S}$ in equation 2 is nearly diagonal.)

INSERT Figure 10: Estimated 90% simultaneous confidence regions for the two criteria ($\overline{\hat{\mu}}$, $\hat{\rho}$) for the four PPCSs

Stage 5: We consider our methodology as a decision support system (DSS); that is, the methodology does not make the final selection of a particular PPCS. Instead, short-term risk (in terms of fill rate) versus long-term costs (in terms of WIP) are presented to the managers so

that they can select a particular PPCS that fits their specific risk attitude.

Figure 10 suggests that Hybrid dominates the other PPCSs: it minimizes both criteria (any reasonable risk attitude implies that managers prefer low WIP, provided the risk is acceptable). Nevertheless, since Hybrid requires the implementation of both Kanban and Conwip, managers might prefer Kanban: the latter is easier to implement in practice, and only slightly increases both criteria values. Obviously, Conwip gives excessive WIP (Conwip has a single control loop), without decreasing the risk of a 'disaster'. Generic gives a WIP that is relatively high compared with Hybrid and Kanban, while it does not decrease risk. However, were risk ignored, then the ranking from best to worst PPCS would be: Generic, Hybrid, Conwip, Kanban; for details we refer to Gaury (2000) and Gaury et al. (1999). *So risk considerations do make a difference.*

What if management cannot accept the fill rate risk quantified in Figure 10? Managers might be prepared to change their threshold value; see Gaury (2000, p. 94). Alternatively, we may add more WIP - by increasing the number of cards of a specific PPCS. This higher WIP (higher $\overline{\hat{\mu}}$ values in Figure 10) may decrease the fill rate risk ($\hat{\rho}$ in Figure 10), but it is more expensive. A final alternative is to try and change the environment such that a lower risk results. Which environmental inputs are important, can be detected through the techniques that lead to Figure 4.

4. Conclusions

In this paper we emphasize that the robustness of the 'optimum' solution is of utmost importance. Yet, most academics and practitioners try to optimize their simulation models for a base scenario only. To incorporate robustness, we propose to add risk analysis; that is, we develop

a methodology that is a stagewise combination of (1) discrete-event simulation, (2) heuristic optimization, (3) risk analysis, and (4) bootstrapping.

We illustrate this methodology through an example on production pull control systems (PPCSs), namely Kanban, Conwip, Hybrid, and Generic. In that study we define robustness as the PPCS's capability to maintain short-term service while minimizing long-term work-in-process, under a variety of scenarios. These PPCSs control a production line with four stations and a single product, originally studied by Bonvik et al. (1997). The conclusion of this example is that risk considerations may indeed lead to the selection of a different PPCS. Selecting the appropriate PPCS may affect a manager's survival of bad times! Therefore we conclude that methods for performance analysis in operations management should account for robustness when recommending a specific PPCS.

Note that our example is merely an illustration. In practice, robustness depends on the real production system and its control system (for example, a proprietary software system), its particular environment (specified through the RA input distribution), its simulation model (is that model validated?), its optimization (does the heuristic search give the true optimum?), and the managers' performance measures and risk attitude.

In future research, our methodology may be improved by optimizing not for the base scenario only. Instead, the derivation of the optimal values for the control parameters should account for the uncertainty of the environmental inputs; also see Mulvey et al. (1995). Actually, this requires very much computer time; see Gaury (2000, p. 97) for an application of this approach to the simplest production control system, namely Conwip. This computer time problem may be solved in the following ways: (i) program the simulation model such that the simulation runs faster (for example, replace Arena by the approach in Hyden and Schruben, 2000), (ii) apply faster optimization heuristics (GAs are notoriously slow), (iii) explore fewer

scenarios, (iv) use more powerful computers including parallel computers and web-based simulation.

Acknowledgments

We received useful comments on an earlier version, from Jan Engel and Freek Huele (Centre for Quantitative Methods CQM' in Eindhoven, the Netherlands), Lee Schruben (University of California at Berkeley), and Kevin Woods (Naval Postgraduate School in Monterey, California).

References

Balson, W.E., J.L. Welsh and D.S. Wilson (1992), Using decision analysis and risk analysis to manage utility environmental risk. *Interfaces*, 22, no. 6, pp. 126-139

Ben- Tal, A. (2000), Robust convex optimization. Fifth International Conference on High Performance Optimization Techniques, Rotterdam

Benjamin, P.C., M. Erraguntla, and R.J. Mayer (1995), Using simulation for robust system design. *Simulation*, 65, no. 2, pp. 116-127

Birge and Louveaux (19..)

Bonvik, A.M., C.E. Couch, and S.B. Gershwin (1997). A comparison of production-line control mechanisms, *International Journal of Production Research*, 35, 3, 789-804

Breeding R.J. et al. (1992), Summary description of the methods used in the probabilistic risk assessments for NUREG-1150. *Nuclear Engineering and Design*, 135, pp. 1-27

Efron, B. and R.J. Tibshirani (1993), *An introduction to the bootstrap*. Chapman & Hall, New

York

Fu, M.C. et al. (2000), Integrating optimization and simulation: research and practice. *Proceedings of the 2000 Winter Simulation Conference* (edited by J.A. Joines, R.R. Barton, K Kang, and P.A. Fishwick), pp. 610-616

Gaury, E.G.A. (2000), *Designing pull production control systems: customization and robustness*. CentER Dissertation Series, ISBN 90 5668 066 8, Tilburg, Netherlands

---, H. Pierreval, and J.P.C. Kleijnen (2000), An evolutionary approach to select a pull system among Kanban, Conwip and Hybrid. *Journal of Intelligent Manufacturing*, 11, no. 2, pp. 157-167

---, H. Pierreval, and J.P.C. Kleijnen (1999), New species of hybrid pull systems. CentER Discussion Paper, no. 9831 (submitted for publication)

Granger Morgan M. and M. Henrion (1990), *A guide to dealing with uncertainty in quantitative risk and policy analysis*. Cambridge University Press

Helton, J.C. (1997), Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty. *Journal Statistical Computation and Simulation*, 57, pp. 3-76

Helton, J.C., D.R. Anderson, M.G. Marietta, and R.P. Rechard (1997), Performance assessment for the waste isolation pilot plant: from regulation to calculation for 40 CFR 191.13. *Operations Research*, 45, no. 2, pp. 157-177

Hyden, P. and L. Schruben (2000), Improved decision processes through simultaneous simulation and time dilation, *Proceedings of the 2000 Winter Simulation Conference* (edited by J.A. Joines, R.R. Barton, K Kang, and P.A. Fishwick), pp. 743-748

Johnson, R.A. and D.W. Wichern (1992), *Applied multivariate statistical analysis*. Prentice-Hall International, Englewood Cliffs, New Jersey

Kall, P and S.W. Wallace (1994), *Stochastic programming*. John Wiley & Sons, Chichester

Kleijnen, J.P.C. (1998), Experimental design for sensitivity analysis, optimization, and validation of simulation models. Chapter 6 in: *Handbook of Simulation*, edited by J. Banks, Wiley, New York, pp. 173-223

--- and J.C. Helton (1999), Statistical analyses of scatter plots to identify important factors in large-scale simulations, 1: review and comparison of techniques. *Reliability Engineering and Systems Safety*, 65, no. 2, pp. 147-185

Krumm, F.V. and C.F. Rolle (1992), Management and application of decision and risk analysis in Du Pont. *Interfaces*, 22, no. 6, pp. 84-93

Law, A.M. and M.G. McComas (2000), Simulation-based optimization. *Proceedings of the 2000 Winter Simulation Conference* (edited by J.A. Joines, R.R. Barton, K Kang, and P.A. Fishwick), pp. 46-49

Lim, J., K. Kim, B. Yum, and H. Hwang (1996), Determination of an optimal configuration of operating policies for direct-input-output manufacturing systems using the Taguchi method. *Computers Industrial Engineering*, 31, no.3/4, pp. 555-560

McKay, M.D., R.J. Beckman, and W.J. Conover (1979), A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21, no. 2, pp. 239-245

Moeeni, F., S.M. Sanchez, and A.J. Vakharia (1997), A robust design methodology for Kanban system design. *International Journal Production Research*, 35, no. 10, pp. 2821-2838

Mulvey, J.M., R.J. Vanderbei, and S.A. Zenios (1995), Robust optimization of large-scale systems. *Operations Research*, 43, no. 2, pp. 264-281

Myers, R.H. (1999), Response surface methodology - current status and future directions.

(Including Discussion.) *Journal of Quality Technology*, 31, no. 1, pp. 30-74

Spearman, M.L., D.L. Woodruff, and W.J. Hopp (1990). CONWIP: a pull alternative to Kanban, *International Journal of Production Research*, 28, 5, 879-894

Sugiyama, S.O. and J.W. Chow (1997), @Risk, Riskview and BestFit. *OR/MS Today*, 24, no. 2, pp. 64-66

Swisher, J.R., P.D. Hyden, S.H. Jacobson, and L.W. Schruben (2000), A survey of simulation optimization techniques and procedures. *Proceedings of the 2000 Winter Simulation Conference* (edited by J.A. Joines, R.R. Barton, K Kang, and P.A. Fishwick), pp. 119-128